Challenges to Bayesian decision support using morphological matrices for design: Empirical evidence

Peter C Matthews[*]

July 2010

Abstract

A novel Bayesian design support tool is empirically investigated for its potential to support the early design stages. The design support tool provides dynamic guidance with the use of morphological design matrices during the conceptual or preliminary design stages. This paper tests the appropriateness of adopting a stochastic approach for supporting the early design phase. The rationale for the stochastic approach is based on the uncertain nature of the design during this part of the design process. The support tool is based on Bayesian Belief Networks (BBNs), and uses a simple but effective information content based metric to learn or induce the model structure. The dynamically interactive tool is assessed with two empirical trials. First, the laboratory based trial with novice designers illustrates a novel emergent design search methodology. Second, the industrial based trial with expert designers illustrates the hurdles that are faced when deploying a design support tool in a highly pressurised industrial environment. The conclusion from these trials is that there is a need for designers to better understand the stochastic methodology for them to both be able to interpret and trust the BBN model of the design domain. Further, there is need for a lightweight domain specific front end interface is needed to enable a better fit between the generic support tool and the domain specific design process and associated tools.

1 Introduction

The fluid nature of the conceptual and preliminary design stages has hindered general design support tools. While there exist specific support tools for certain design domains, these have been constructed at considerable expense due to the need to acquire and encode domain specific knowledge into the support tool. A further issue with support tools for the earlier phases of the design process is the inherent fluidity of the design at this end of the process. Specifically, it is difficult to provide objective metrics to measure the quality of design concepts. As design concepts still require significant effort transform into the final product, there is the potential for a 'good' concept to be poorly detailed and thus result in a poor final product and *vice versa*: a 'poor' concept can be carefuly developed through the detailing phase to result in a 'good' final product. The terms 'good' and 'poor' in this case are context dependent, and cover such criteria as technical quality, commercial success, and aesthetics. However, in general good concepts are more readily transformed into good final products whereas poor concepts require greater effort to ensure a similar final high quality level.

For this paper, conceptual design will be defined as the point in the design process when the designers have transformed the initial product specification to a form that defines the nature of all key elements of the final product (Pahl and Beitz, 1996; Cross, 1994; Ulrich and Eppinger, 2000). This paper presents a

[*] School of Engineering and Computing Sciences, Durham University DH1 3LE; phone +44 191 334 2538; email: p.c.matthews@durham.ac.uk (corresponding author)

Bayesian approach to supporting the decision process in the conceptual design stage. Central to implementing the Bayesian approach is the need to acquire good stochastic information about the design domain. This is achieved through the use of databases of prior design examples. These databases of conceptual designs require defining, and hence there is a need to define conceptual design. In this context, conceptual design will be considered in a morphological sense: the design will be structured into a set of functional and characteristic variables (Pahl and Beitz, 1996; Hollins and Pugh, 1990). Against each of these variables, a designer will have to select a conceptual solution. The combination of all these solutions then generates the final design concept. While this approach has traditionally been used to represent different solution categories, in this work this will be simply extended to also include the representation of numerical design parameter values as well, for example the overall wing span of an aircraft. This representation enables this methodology to be used in design domains where some aspects of the design require such values. Ultimately, generating 'good' design using this approach relies on the designer understanding the interaction between the design variables.

There have been other attempts at supporting the conceptual design stage. Most of these attempts implement a rule based system, that is specific domain knowledge is encoded within the support system. A typical example of such a support system is given by Zhang et al. (2002). This system takes a functional/behaviour based approach. The system incorporates a database for matching desired behaviours to known functions. However, the rules required to encode these functions are complex, although no specific details are given within the paper to the cost associated with encoding the rules. Design concepts are then scored according to their simplicity as determined by the part count required to instantiate a given concept. It then remains the designer's decision which concept to select. Another example of a rule based system is provided by Geyer (2008). This is a more complex approach based on shape grammar. The shape grammar provides a framework for being able to decompose and subsequently recompose a design while maintaining functionality. These actions are based on a set of predetermined rules. By applying these actions, a set of design alternatives can be automatically created and presented to the designer for final decision. In conclusion, while rule based systems are able to automatically perform design search, they present the designer with a small set of fully determined design alternatives from which one must be selected. Although the designer has flexibility in which alternative to select, the presented alternatives are rigid. A further example of conceptual design support is given by Ziv-Av and Reich (2005) which provides a means for generating optimal concepts subject to the designer's subjective objectives. This approach decomposes the design problem to support the design search process. The designer is ultimately presented with a rank ordered set of design concepts to take further.

In contrast to a rule based approach, adopting a stochastic view of the design domain results in the fluid nature of the early design stages becomes relatively simple to represent. As the underlying model is stochastic rather than rule based, multiple outcomes are natively represented. There are a number of important benefits to be gained from using a stochastic view of the design domain (He et al., 2006). First, a stochastic approach natively supports missing or imprecise input data. This degree of uncertainty is common during the conceptual design stage, as not all aspects of the design are likely to have been determined. Second, a stochastic approach is able to perform inference under uncertainty. Perfoming inference under uncertainty is much more challenging for rule based systems, as the uncertainty of the input data means that the rule based systems are unable to trigger production rules and thereby provide useful feedback to the designer. Finally, the stochastic approach is also able to provide stochastic inference. This inference supports multiple outcomes, which is a significant difference from the rule based approach. Under a rule based approach the outcome is deterministic, that is given the input there is a unique output. Where alternative solutions are provided, these tend to be as a result of a search process and again are deterministically scored. However, under a stochastic approach multiple alternatives can be presented for a given input. These alternative outcomes are then presented with an associated probability for each outcome. Therefore, at any point in the early design process, the range of potential outcomes of any decision can be represented using probability distribution functions (PDFs). The most likely outcome is represented by taking the maximum value of the PDF, but the designer is clearly presented with the other potential outcomes. Assuming that the PDFs represent the probability of an outcome being present in the final design, a designer can get a feel of the likelihood of a successful final design based on the decisions made. Therefore, the stochastic support tool effectively suggests that the designer should follow the path of greatest probability but allows other paths to be explored should the designer wish to. By following a low probability path, the designer understands that greater care must be taken later in the design process, as this path is one that has not been successfully followed previously. This paper presents two empirical trials that seek to test if these benefits are realisable.

This paper introduces a stochastic design support tool based on Bayesian Belief Networks (BBNs). A brief overview of the underlying theory will be presented (Section 3). Greater attention is given to the empirical assessment of this design support tool (Section 4). The empirical work leads to a critical discussion on the nature of developing and implementing generic design support tools (Section 5).

2 Background

Design support tools have been an active area for some time. Research in the development of support tools tends to be focussed on supporting one type of design activity or a particular design domain. Chong et al. (2009) report that a good design theory is required prior to any attempt is made on providing computational design support. To underpin the stochastic support tool described in this paper, the morphological matrix design framework is adopted (Pahl and Beitz, 1996). The morphological matrix provides flexible support, yet without requiring any rule base to be put into place prior to using it. The support tool that is presented in this paper has the potential to provide computational support for efficient use of the morphological matrix during the conceptual design phase.

An important aspect of conceptual design is enabling the designer to fully search the design space. One flexible method of supporting this is the use of morphological matrices. While this approach enables the designer to visualise and construct a very wide variety of potential designs, it provides no support in providing feedback to the designer about the likelihood of a particular combination (or partial combination) resulting in a successful design. This section will consider previous design support tools, and what contexts these different tools operate within. These then inform the support method for morphological design that are developed in Section 3.

To be able to successfully develop a design support tool, there are four aspects that need to be adressed: (1) morphological matrix definition, (2) design model structure, (3) knowledge acquisition, and (4) interfacing the model with the designer.

2.1 Morphological Matrix

One common method for spanning and exploring the conceptual design space is the morphological matrix (Zwicky, 1967; Hollins and

Pugh, 1990; Pahl and Beitz, 1996). The design space is spanned by the set of functional modules of the product, that is the set of independent functions or properties that are expected of the product. For example, consider the (very simplified) design of an aircraft. An aircraft can be described by the following: the primary construction material, the propulsion system, the capacity, and the wing geometry. Against each function, there are a number of possible solutions. Considering the propulsion system function, there are four possible soltions: jet, turbo-propellor, piston-based propellor, and none (in the case of a glider). Figure 1 expands this aircraft example, providing a more complete functional breakdown of the conceptual design space. The design space can be explored by selecting one solution against each functional module of the product. By selecting a functional solution for each functional module, a designer can then construct a full design concept. Figure 2 illustrates four separate concepts that can be extracted from this particular morphological matrix example. Finally, it is worth noting that there is no need to restrict the solutions to discrete nominal categories: it is possible to also have continuous solutions, for example the capacity could have been represented by the number of passengers using the integer range 1–1000.

The morphological matrix provides a means for visualising the conceptual design space and composing potential design concepts. However, there are a number of challenges associated with this approach. The first is the explosion in number of potential concepts that can be created from the matrix. As in theory the selection of each function solution is independent of all other selected function solutions, the theoretical total number of concepts that exist is given by the product of the number of solutions for each function. In the case of the aircraft example in Figure 1, there are $4 \times 4 \times 4 \times 3 = 192$ different concepts. A significant number of these will be, if not physically impossible, highly unlikely, for example any combination of a cloth body and a jet engine. Clearly, the choice of function solutions are not independent of each other. This leads to the second challenge: what is the best order in which to determine function solutions? Depending on the design specification, a rigid ordering is unlikely to be helpful. Different design specification will predetermine some function solutions and therefore the designer will set the remaining function solutions in a different order. Finally, as the morphological matrix is a static representation, no guidance is provided on the suitability of different function combinations. Instead, a designer can only rely on their intuition and tacit knowledge about the domain when combining function solutions.

2.2 Design model structure

The design model structure defines not only how the model is represented, but also determines to a large extent how the computational support tool operates. Historically, the design models and support tools adopted a case based approach. Prior designs are encoded and entered into a database so that when a designer comes to create a new design, previous similar designs are presented (Maher and Gómez de Silva Garza, 1997; Rivard and Fenves, 2000). An extension of the case based approach is the knowledge based design support. With this system, engineering knowledge is stored within the design support system. This knowledge is formatted as a set of rules that can be applied to the evolving design. For example, Moore et al. (1997) use an object oriented representation of both the design and knowledge base to provide support in the form of design critique. When the system notes that a designer is either potentially in error or the system identifies a potential alternative solution, this is brought to the designer's attention. Scwabacher et al. (1998) use a machine learning approach to support a designer in setting up an optimiser, along with predicting design goals. On a similar

Function	Solution			
Material	Carbon Fibre	Aluminium	Fibreglass	Cloth
Propulsion	Jet	Turbo−prop	Propeller	(none)
Capacity	Single	Small	Medium	Large
Wing Geometry	Thin Chord	Medium Chord	Thick Chord	

Figure 1: A morphological matrix for a conceptual aircraft design space.

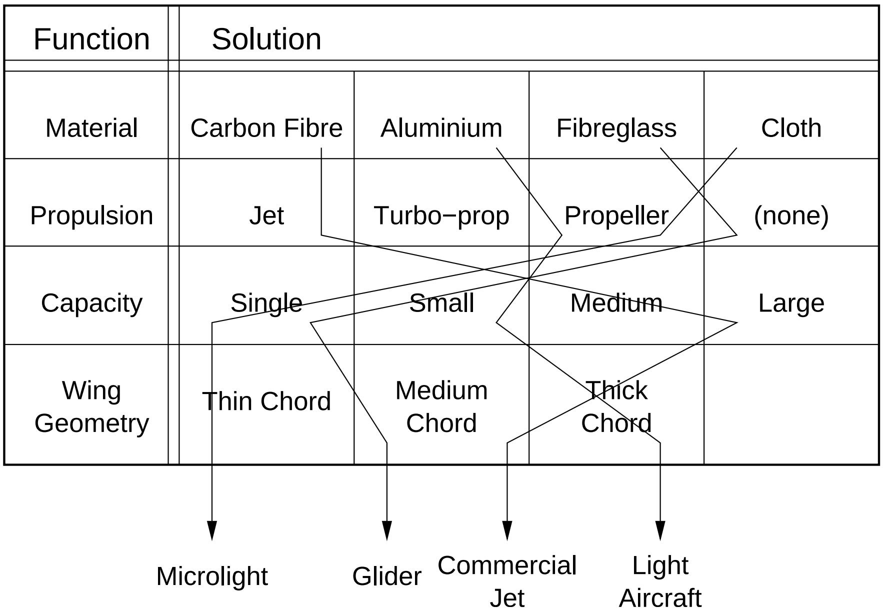
note, Ong and Keane (2002) provide a support method that advises a designer on suitable optimisers to use for a given design problem. This support was provided by representing the optimisers according to their characteristics, thus forming a knowledge base about design optimisers.

In addition to case and knowledge based representations, stochastic approaches for representing the design domain have also been used. An early report of using a Bayesian approach described modelling an electronic chip design domain (Osio and Amon, 1996). Computer simulations were used to generate training data to create the probability distributions, and this enabled a more cost effective modelling of the design domain. More recently, Simpson et al. (2001) and Chen et al. (2006) discuss the creation of statistical meta-models of design domains. These focus on the two aspects of meta-modelling, namely the selection of a suitable statistical model class and then the fitting of the model to the empirical evidence. One of the benefits of using a stochastic approach is that it becomes feasible to model non-deterministic aspects of the domain. There are a number of examples of using stochastic approaches to model domains with strong 'external' events, such as flood risk (Apel et al., 2006) or subjective evaluation and the resulting behaviour as exhibited by the spreading of subjective cinema evaluations by the word of mouth (Eliashberg et al., 2000).

The robustness of a design represents a particularly interesting non-deterministic characteristic of a design that can be represented stochastically. Robustness has been stochastically considered in the context of uncertainty in planning (Sahinidis, 2004) and in change propagation (Clarkson et al., 2004). There has also been work in categorising risks and using this to indicate promising design directions (Pons and Raine, 2004, 2005). Non-stochastic approaches to risk modelling include inducing regression trees to identify pareto-optimal designs (Forouraghi, 1999) and using utility function based approaches for risk mitigation based on designer preferences (Fernandez et al., 2005).

2.3 Knowledge acquisition

Given a model of a design domain, this is of little use beyond design representation without any knowledge about how different aspects of the design domain interact with each other. This knowledge must be acquired: either deduced from physical laws, from designers in person or induced from analysis of previous designs.



Function	Solution			
Material	Carbon Fibre	Aluminium	Fibreglass	Cloth
Propulsion	Jet	Turbo−prop	Propeller	(none)
Capacity	Single	Small	Medium	Large
Wing Geometry	Thin Chord	Medium Chord	Thick Chord	

Figure 2: Four aircraft concepts derived from the morphological matrix.

This knowledge acquisition is a known bottleneck for developing expert systems (Gaines and Shaw, 1993), and by extension design support tools. Methodologies have been developed to acquire design knowledge from domain experts, but this tends to be highly tailored to the particular domain under review (Hughes et al., 2001). A more automated approach is to use machine learning techniques to support the knowledge acquisition process. This typically involves analysing a set of design examples to induce a domain model (Potter et al., 2001). There are a number of examples and approaches to this methodology. Pacheco et al. (2003) use a covariance based method to generate a rough surrogate model that can provide a designer with a basic understanding of the domain relationships. Mahdavi and Gurtekin (2004) use a neural network to generate a design performance space that can then be visualised by a designer to provide greater insight into the design performance space, thus enabling a designer to explore the relationships between sets of design variables. Matthews et al. (2006) use an augmented genetic programming method to identify a set of relationships that provide an algebraic description of the relationships between the design variables.

2.4 Designer interaction

The final aspect to consider when developing a design support tool is the interface between the design model and the designer. This represents the most challenging part of the design support tool, as a good user interface must consider the fundamental human functions and tasks related to the design domain (Rinkus et al., 2005). There is relatively little technical literature on this topic. Most of the literature concerns generic user interface development. The research tends to focus on general user interface development methodologies, for example best practises (Szewczyk, 2003; Resnick and Vaughan, 2006) and practical reports (Kim and Yoon, 2005).

The research in this paper does not extensively consider the user interface, beyond using software development and implementation tools that ensure wide access to support active designers. Further details of the user interface are presented in Matthews (2008).

2.5 Summary

From the literature, there are four key areas relevant to this research. The first two relate to the mode of knowledge representation: rule based versus stochastic based. These two modes compete, and this research seeks to empirically demonstrate the potential benefits of the stochastic approach. The argument against the rule based approach is two-fold. Firstly, there is the complexity of the rules required to populate a useful rule based system. Secondly, there is the rigidity of the application of the rules. The designer is constrained to apply the rules as presented, rather than to be able to slightly modify the outcome to one that the designer believes to be more appropriate. These issues are overcome through the adoption of a stochastic approach. The designer gains flexibility, and is informed through the related probability values of the chance of success.

The third area relevant to this research is the design domain knowledge acquisition. This typically is a bottleneck area, and attempts have been made to use machine learning techniques to streamline this area. Finally, the fourth area is the interface between the design support system and the designer. This area is unfortunately one with the least literature specific to design support.

3 Stochastic modelling approach

The stochastic approach to modelling an uncertain domain enables a natural representation of the fluid characteristics inherent with uncertainty. This is in contrast to rule based modelling approaches, where outcomes are deterministic. The design concepts generated using morphological matrices require further detailing and hence contain uncertainty. There are ample opportunities to modify these designs further downstream, albeit less significantly than is possible during the earlier stages.

The challenge of creating a suitably representative domain model remains. The model must represent the relationships between the various design variables, specifically how changes to one variable are likely to affect other aspects of the design. There are two main approaches to obtaining these relationships: acquiring the knowledge directly from domain experts or analysing design databases using machine learning techniques. Due to the relatively high cost of the knowledge acquisition from domain experts, this research focuses on investigating the use of machine learning algorithms.

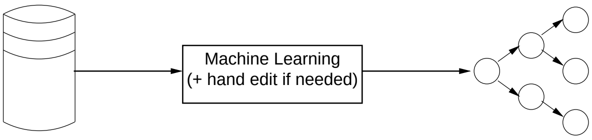


Figure 3: The overall machine learning process: the design database is used to induce a Bayesian Belief Network representing the design variable causal structure.

Bayesian Belief Networks (BBNs) have been selected to represent the domain model, as these provide a simple and relatively intuitive perspective on the domain. These models are created using an information content based metric. Figure 3 illustrates the machine learning process for creating the BBNs. In larger domains, even BBNs can become complex. Therefore, an additional aspect of this research is to investigate the potential of breaking these networks into smaller domain sub-space models that are more readily understood. Each of these aspects is expanded on below.

3.1 Bayesian Belief Networks for Design

The design space is encoded parametrically. The functions from the morphological matrix that span the design space define the parameters and the function solutions represent the values that each parameter takes under this design encoding scheme. These variables are interpreted as ‘observations’ in the BBN sense. The BBN is a graphical model, where the nodes (design variables) are linked by directed arcs. These directed arcs represent the causal dependencies between design variables. Where a design variable is 'observed' (i.e. has a value associated to it), it becomes possible to infer the likelihoods and/or probabilities of the values that would be taken on by neighbouring variables. Figure 4 illustrates the process of using the BBN as part of the conceptual design process.

A key difference with this approach to modelling the design space is that both design parameters (aspects of the design that are directly determined by the designer) and design characteristics (aspects of the design that are the result of the designer's choices) are both represented as abstract design variables. The approach taken in this research removes this distinction and allows the designer to specify either design parameters or characteristics at the outset. This is a powerful abstraction, as a designer is not constrained to first determine the design parameters but can determine the characteristics and then use the tool to be provided with guidance on how to achieve these design characteristics. This approach of removing the parameter–characteristic ordering is valid. Consider a hypothetical aircraft design process: design parameters include the wing geometry, the skin material, and the propulsion type. Design characteristics include the lift and drag coefficients for the aircraft. The choice of material will affect the drag coefficient which in turn will affect the choice of wing geometry and propulsion system. If all the design parameters are considered first as a group, this results in the need for an iterative approach as the design characteristics that do not meet the specification can only be affected by modifying the design parameters. This process is then repeated until a successful design is achieved.

The stochastic nature of BBNs makes this a highly flexible approach to modelling the design space. Conceptual design is by its nature very fluid: any given design concept can be detailed later in the design process in a number of different ways. This leads to the following argument: a 'good' design concept has a high probability of becoming a 'good' final product, while a 'poor' design concept has a low probability of becoming a 'good' final product. The stochastic representation enables the opposite outcome in both events, however these are represented by accordingly low probabilities. Further details of this illustration are provided by Matthews (2008).

3.2 Information content based learning

There are two key aspects to the BBN: (1) the acquiring of the distribution functions and (2) the creation of the variable network. It is assumed that a database of prior designs is available. This database is populated with the parametric and characteristic values taken from previous designs, and could also include the results of costly emprical and computational results. Using this database, it is a relatively trivial task to generate the conditional distribution function, given a network. The second aspect, creation of the network representing the causal relationships within the domain, is a considerably more challenging task. Where no information about the causal ordering is known, this is theoretically a very computationally expensive task (Pearl and Verma, 1991; Pearl, 1995). An alternative method is to obtain the network structure from domain experts. While this is computationally much simpler and does not require a design database, it does place a very large task on the domain experts. As the number of design variables increases, the amount of effort required by the domain expert in considering possible relationships increases quadratically. Hence, for realistic design domains, this is not an appealing method.

A computationally more efficient approach has been adopted based on the information content that would be contained in any potential causal arc (Matthews, 2006, 2008). This approach is similar to that developed by Chen

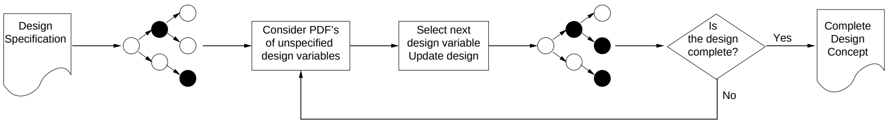

Design Specification
Consider PDF's of unspecified design variables
Select next design variable Update design
Is the design complete?
Yes
No
Complete Design Concept

Figure 4: The design process using the BBN starts with the design specification being entered into the BBN (filled nodes). The conceptual design is then iteratively completed by the designer, guided by the variable PDFs.

et al. (2002) which measures the conditional independence between pairs of variables. Where the causal ordering of the variables is not known in advance, they demonstrate that it is possible to learn a suitably accurate network complexity of $O(n^4)$ in terms of the number of variables. The approach described in Matthews (2008) uses a greedy algorithm that learns a network with complexity $O(n^2)$. This approach uses the basic definition of conditional probability:

$$\mathrm{P}(B = b \,|\, A = a) = \frac{\mathrm{P}(B = b, A = a)}{\mathrm{P}(A = a)} \qquad (1)$$

Where the events, or variables, A and B are independent, $\mathrm{P}(B, A) = \mathrm{P}(B)\mathrm{P}(A)$. Therefore, when A and B are independent, $\mathrm{P}(B|A) = \mathrm{P}(B)$. By considering the difference between the observed conditional and prior probabilities taken from the design database, it is possible to measure the information content that is contained within this conditional probability distribution. Hence, the following information content metric is defined between variables A and B:

$$I(A, B) = \mathbf{E}[\mathrm{P}(B \,|\, A) - \mathrm{P}(B)]^2 \qquad (2)$$

Where the variables are truly independent, the measured difference will be zero, and hence a zero information score will be returned. This indicates that there is nothing to be gained by including this arc. Where a large I value is returned, this indicates that the conditional probability distribution contributes significantly to the domain knowledge and hence this arc should be included. It is also worth noting that this information metric is asymmetric, namely $I(A, B) \neq I(B, A)$. This is useful, as it provides an indication of the causal direction of relationships within the network.

This information metric is measured for all variable pairings in both directions. For each variable, the arc with maximum information content either entering or leaving that variable is identified. These arcs form a set of partial models. Next, the two models that share a variable and have the lowest total information content are merged, to form a new partial model with three variables. This process is continued until all nodes are connected, resulting in a full model for the domain. The flowchart representing this process is given in Figure 5.

4 Empirical work

To verify the effectiveness of a Bayesian design support tool, two empircal studies were undertaken. The two studies were largely complementary in their nature, rather than reinforcing. The first used the support tool within a laboratory setting, enabling a fine grain of control of both the design domain and methods used by the designers. In this study, the user interface and the representation of the design model was not under investigation. The aim was to measure how well the BBN induction algorithm was and how effective the decision support tool was when no alternative was available. The second empirical study was undertaken within an industrial setting. This provided a real life design problem, three times larger than the laboratory based study in terms of number of design variables, with

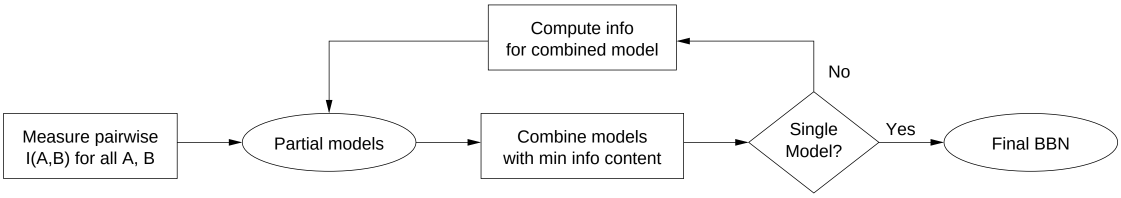


Figure 5: Flowchart representing the greedy BBN learning algorithm.

all the complexities and subtleties that this entails. This, however, was at the cost of control over the experiment, and the results from this study were of a more subjective nature. In this study the aim was to determine how acceptable the BBN design domain representation was, from the perspective of domain experts. This provided a critique on the Bayesian design domain representation and of the effectiveness of the user interface.

Each case followed the same process to induce a BBN. First, the design data was acquired and where necessary discretised. This data was then processed by the structure learning algorithm to produce the BBN network. This network was then imported into the design support tool, for interactive use. At the same time, a static hard copy of the BBN structure was made available for designers to refer to while using the tool. This provided them with a representation of the causal relationship structure between the design variables, thereby enabling the designer to understand how changes in variables would affect the probability distributions in other variables. Each case study was followed by an analysis on the usability and usefulness of the support tool within each context.

4.1 Laboratory based study

The laboratory based study used the 'Car Design' machine learning benchmarking database from the University of California at Irvine (Blake and Merz, 1998). Although this is an artificially constructed database, it does represent a good approximation to a real conceptual car design database. Additionally, the variable structure is also known, thereby enabling a direct comparison between the induced network and the actual deterministic network. The scope of this paper presents the qualitative aspects of this study that inform the specific challenges of deploying a stochastic design decision support tool. An in-depth quantitative analysis of this study alone is presented in Matthews (2008).

The car database contains a sample of 1728 fully described designs. The design domain is defined by ten variables, of which six are design parameters (the target purchase price; the expected maintenance cost; the designed safety level; the number of doors; the number of passengers; and the volume of luggage that can be carried) and the remaining four are design characteristics (the overall cost of ownership; the comfort level; the technology level; and the overall car acceptability). All the variables are discrete, and hence this fully defines the domain's morphological matrix, and a more detailed description of the variables is listed in Table 1. The original car design database was constructed using a set of predetermined rules. The structure of these rules is provided in Figure 6. This known rule structure makes it possible to evaluate the quality of the machine learnt domain model.

The car database was first loaded into Matlab and passed to the BBN learning algorithm. This generated a network representing the causal links between the design variables. The algorithm identifies exactly as many arcs as there are design variables. This resulted in a non-tree structure. In a tree structure each node, with the exception of the root node, should have a single child. The structure that was

Table 1: Car design morphology table: variable names, abbreviations, and description. The descriptions include the possible variable values. Design parameters are in lower case and the design characteristics are in upper case.

Name (abbreviation)	Description
buying (buy)	Purchase price for car (low, medium, high, very high)
maintenance (mnt)	Expected maintenance cost for car (low, medium, high, very high)
doors (drs)	Number of doors on car (2, 3, 4, 5+)
persons (pers)	Number of passengers (2, 4, 5+)
luggage (lug)	Available luggage volume (small, medium, big)
safety (safe)	Designed safety level (low, medium, high)
COMFORT (CMFT)	Comfort level of car (unacceptable, acceptable, good, very good)
PRICE (PRC)	Total cost of ownership (unacceptable, acceptable, good, very good)
TECHNOLOGY (TECH)	Technology level of car (unacceptable, acceptable, good, very good)
CAR (CAR)	Overall acceptability of car (unacceptable, acceptable, good, very good)

produced by the learning algorithm had the 'safety' node linked to both the 'technology' and 'car acceptability' nodes. By considering the information content of these two arcs coming out of the safety node, the arc with the lower information content was deleted which in this case was the arc leading to 'car acceptability'. The impact of this deletion is minor, as 'safety' is still related to 'car acceptability', albeit indirectly through the 'technology' node. The resulting tree network that was learnt from the dataset had an identical causal structure to the underlying rule structure used to create original the design database, as illustrated in Figure 6. Subsequently, this network was encoded in the Excel spreadsheet, along with the design database.

The stochastic approach was compared against a more 'traditional' approach to design. This required the designers to consider the design characteristics separately, without any information about feasibility of the design until the design had been completely specified. This approach was implemented using a similar interface to ensure that a fair comparison was possible between the Bayesian and traditional design approaches.

4.1.1 Summary of the effects of Bayesian design

The car design case study provided in this section illustrates three key aspects of the stochastic design support tool. The first is that the machine learning algorithm induces a suitably good domain model from a set of prior design examples. The algorithm produced a graph structure with one arc too many for it to be a tree structure, as required for a BBN. A tree structure is required, as the implementation of this code is only able to handle one input arc per node. By using the information content heuristic, it was possible to identify which arc should be deleted and this resulted in the same structure that generated the data in the first instance.

The second aspect of the stochastic support tool is that the design search process can begin with a partial design specification. This was demonstrated by starting the de-

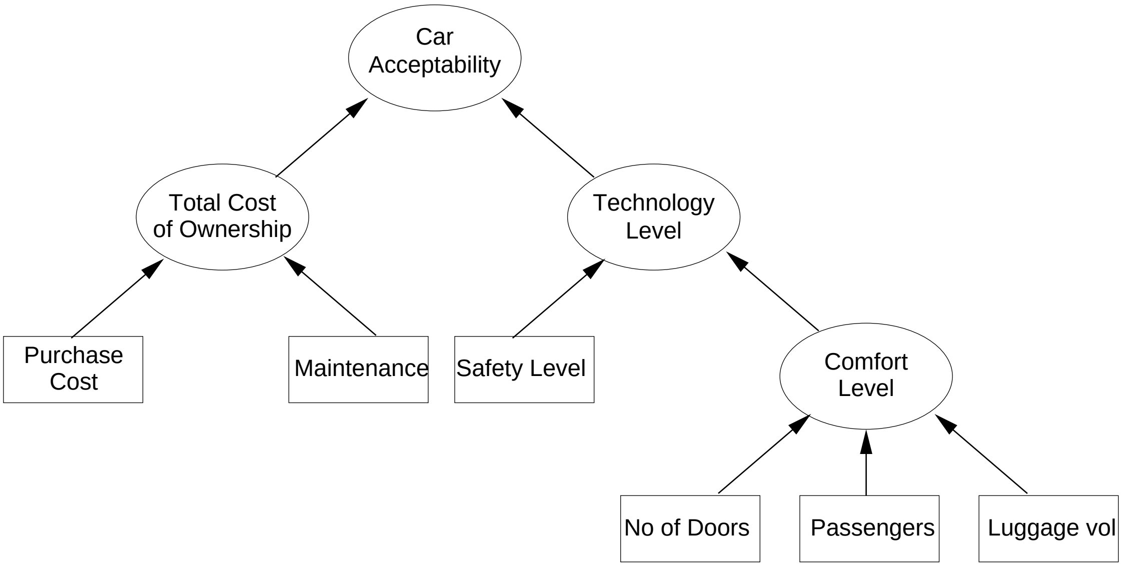


Figure 6: Rule structure for the conceptual car domain.

sign search with a specification on a subset of both design parameters and characteristics. The interactive search tool then guided the design refinement process, variable by variable. For each variable, the various possible settings were ordered according to the probability of a successful outcome. The designer is encouraged to follow this 'path of greatest likelihood', but is not compelled to. The illustration of this search process thus provides evidence for the third key aspect, namely that the design search heuristics lead to an efficient yet flexible design search path.

4.1.2 Empirical comparison of the two design approaches

The stochastic and traditional approaches were empirically compared. This involved primarily recording how designers used these two different approaches. For this experiment, undergraduate students were used as subjects. These subjects had limited experience of design, and were therefore open to exploring new approaches. All subjects were given training in the design approach they were going to use as well as how to use the design support tool interface. Throughout the design search process, observations were made on how each designer used their support tool. This observation was primarily to evaluate the effectiveness of the user interface. The full results are reported in Matthews (2008). The key points of this trail are summarised here, with a focus on the benefits and challenges that were observed.

The results of this trial were interesting in two ways: (1) there was not a statistically significant difference in the performance between the two tools, but (2) the designers using the stochastic support tool developed a novel search *approach* to the design space that was not expected. The resulting designs and time required to create those designs was very similar across the two different tools. A small number of design variables were statistically different, but not enough to provide significant evidence that the two tools result in different designs, or that one tool results in designs being generated significantly faster than the other. However, the second point illustrated that the stochastic tool lead designers to explore the design space in a novel manner. Both tools were used in a 'test' mode to explore the design space: designers would enter speculative design variable settings and see what effect that would have on the remainder of the design space. In the rule based tool, this effect was seen on the design characteristics as a result of modifying a design parameter. The stochastic tool provided information more broadly: a designer would set a particular design variable, and then see how this would affect the various PDFs throughout the remaining unspecified design. Effectively, they were monitoring a far greater number of variables with greater breadth within each variable than is possible with the rule based support tool. This represents a significant advance in the ability to search a design space interactively.

The reason for the lack of any significant difference between the two support tools can be in part explained by the simple nature of the design domain. The conceptual car domain that was used is relatively intuitive. Therefore, it can be expected that the subjects using the rule based tool were readily able to get a rough feel for the domain rules. The subjects using the stochastic support tool on the other hand needed to learn the how to interpret the displayed PDFs. It is not as clear if these subjects gained an equivalent 'feel' for the design domain as the rule based tool trial subjects were able to. However, the stochastic support tool subjects did develop a novel search methodology that reveals at any point a wider set of options than is possible using the traditional design support tool.

4.2 Industrial based study

The industrial trial was undertaken with a large aerospace company that designs and manufactures a large range of gas turbine aeroengines. The focus of this trial was centered around the combustor system within the gas turbine. The gas turbine can be abstractly thought of having three key systems: (1) the compressor, which takes the gas from the outside at ambient pressure, raises the pressure of this gas which is then passed to (2) the combustor where the gas is mixed with fuel which is then combusted to heat the gas before it continues on to (3) the turbine which extracts mechanical energy from the gas stream. The combustor has a challenging task: the incoming high velocity and pressure gas stream must be slowed down to a speed in which a flame can be sustained, the flow within the combustor needs to be sufficiently turbulent to achieve good fuel mixing, and a film of cool air on the combustor wall is required to protect the combustor from the high flame temperature. The combustor is effectively a cylinder, and the gas from the compressor is passed along the outside of this cylinder. The gas then enters into the cylinder through a series of holes on the surface of the combustor. It is the designer's job to determine the type and placement of these holes on the combustor cylinder. Due to the turbulent nature of the gas flow within the combustor, it is very costly to obtain accurate prediction to the behaviour of the gas for a given combustor configuration. It is the difficulty in obtaining rapid design feedback that provides a key challenge to the designer. The aim of the research was to investigate the potential of a rapid stochastic design decision support tool to guide the designer through the selection and placement of holes on the combustor cylinder.

4.2.1 BBN Induction

A design database of preliminary gas turbine combustors was used to induce the Bayesian Belief Network and provide the underlying probability distribution functions for all the design variables. The database contained a sample of previous combustor designs that had been created by the designers, and the data included various evaluations for these combustors. The combustor database represented a considerably larger design domain than the car design laboratory based study. Further, unlike the car domain, the causal relationship structure between the design variables in this study was not explicitly known. Therefore the induced causal structure and subsequent design support tool were to be critically reviewed by a team of the company's senior combustion design experts.

The combustor design dataset contained a combination of both continuous valued (e.g. pressure) and discrete valued (e.g. hole type) design variables. The continuous variables were discretised into seven categories. Two discretisation approaches were used: a uniform discretisation, where the discretisation boundaries within each variable were kept equally distant between each other; and non-uniform discretisation, where each discrete category had the same number of samples within it. The two discretisation methods resulted in a slightly different BBNs, both of which were considered by the combustor design team.

As there were 25 design variables, the combustor design domain represented a significantly more challenging representation problem that the car domain. To handle this higher complexity level, the structure learning algorithm identified a set of smaller, independent, BBNs. The same structure learning algorithm was applied to the combustor data set. The direct result in this case was again not a tree, specifically a number of variables had multiple arcs coming out from the variable node rather than one as would be the case in a tree. This was rectified again by considering which arcs could be deleted to transform the network into a tree and then eliminating those arcs with the lowest information content. This resulted in creating a set of five trees, with each tree representing a sub-space of the whole design domain.

4.2.2 Experimental Parameters

The industrial experiment focused on how well the stochastic design search tool supported industrial designers in the early design stages. The induced trees representing the causal relationships within the combustor design domain were presented to the industrial design team. This presentation included a brief tutorial on how to use the stochastic support tool. The design team were also left with a written overview of both the underlying theory, the BBNs, and a user guide for the interactive designer support tool. The design team were then left with the support tool for them to trial in their own time. In addition to the tool, the design team were asked to consider the following criteria to assess the support tool against:

1. How well does the tool support the design process;
2. How well does the model match your understanding of the design domain; and
3. How could the user interface be improved.

After the design team had a chance to investigate the support tool, they had difficulty in determining how well the stochastic decision support tool was able to merge with the combustor design process. The combustor design process involves building a network of gas flows which is then incrementally adjusted using a flow analysis tool and the designer's expertise. The initial network can be taken from a previous design that meets a number of the new design's requirements. Although there is a clear morphology for design elements within the combustor design, these apply at a very local level. It was not clear how the stochastic design tool should support the designer in this case.

4.2.3 Results and Discussion

The nature of the stochastic design support tool was insufficiently aligned with this stage of the combustor design process. Specifically, there were three principal reasons that the support tool was not adequate, in response to the above criteria that the designers were asked to consider:

First, as the designers would typically consider different configurations separately, they were expecting separate explicit models for each configuration. The stochastic support tool was not able to support this global view of the combustor model. Rather than a global combustor model, it considered the impact that a given hole type and geometry would have locally on the gas flow. Without the global combustor aspects being made explicit, the designers could not be confident about assessing the accuracy of the single configuration model.

Second, the support tool was trained on a series of different combustor configurations. As a result of the machine learning process, the designers felt that there were a number of aspects within the induced causal model that did not make physical sense according to their understanding of the combustor domain. It was not clear if this represented an unexpected aspect of the combustor domain that the designers were not aware of, i.e. that the machine learning process had discovered novel aspects of the combustor domain, or if this was due to the causal model either extrapolating to regions it should not do or averaging what should be different models into an incorrect combination of models.

Third, as a result a lack of clarity in the presentation of the model, the designers did not feel they were able to provide constructive criticism on the nature of the user interface. The fundamental problem here was that the designers did not feel comfortable with the stochastic approach to decision support. The stochastic approach, as discussed in Section 1, allows multiple outcomes, weighted by their probabilities, to be simultaneously possible. The designer then selects which of these outcomes they would prefer, guided by the PDF. The presentation of this information and the understanding of the impact of a choice requires the decision maker to have a good understanding of the underlying stochastic theory. The fundamentals of the stochastic decision theory were introduced as part of the trial, it is clear that not enough time was devoted to ensuring the designers had a good grasp of this approach.

Overall, the design team could not reach a consensus on the benefits of the support tool. It was felt that either the tool should be deployed at an earlier stage of the design process where there is more flexibility in the decision process or that the decision support tool should be able to integrate a number of different models where obtaining trade-offs between design variables is difficult. These are both aspects that currently require designer expertise and have not been formally captured. The training data for the stochastic support tool was obtained from an intermediate combustor evaluation tool, and therefore did not represent one of the more challenging aspects of the combustor design process. Obtaining training data for the other aspects of the design process is a challenge, and was beyond the scope of the collaboration in this project.

In summary, the key challenge that arose through this process is that the tool must be intimately developed with the designer's processes and needs. Further, and significantly for this approach, the data that is used by the machine learning algorithm must be directly linkable to the data that the designer directly uses and manipulates. In this case, the data used for the machine learning process was taken from an intermediate process that is not directly manipulated by the designer. This resulted in the designer not being able to intuitively sense if the support tool was providing meaningful and useful support to the design process. Ultimately, the designers were not able to successfully consider the design support tool against the three criteria set out in the previous section.

5 Discussion

The empirical studies conducted highlighted three challenges affecting the implementation and adoption of a general stochastic design support tool. First, it is important to select and fit an underlying domain model. This provides the computational foundation for the support tool. Second, the model, and by extension the support tool, must be perceived to be of a sufficiently high accuracy as needed for the design task at hand. Finally, the tool must be accessible by a designer. Specifically, the benefits of using the tool must outweigh the cost of learning to use the tool.

When selecting the underlying model, it is critical to consider how it will approximate the design domain. There are two options at this point: it can either be explicitly provided with domain rules or it can use previous data as a means for model fitting to the domain. Providing the domain rules explicitly requires these rules to be acquired, this represents an expensive approach in terms of the time required by a knowledge engineering to extract these rules (Potter et al., 2001, 2003). The alternative approach is to use machine learning algorithms to fit domain models based on prior design examples. Provided the design data is readily available, this removes the cost of using a knowledge engineer.

In the laboratory case study, both rules and data were available, allowing for the induced model to be verified against the original domain rules. The induced model compared favourably against the original domain model, providing evidence that the machine learning algorithm was sound. There was the need to delete a single arc to maintain the tree structure. In general, this does not pose a problem: the deleted arc is the one with the lower information content. From a knowledge representation perspective, the arc would either be linking the node to a node higher in the tree (bypassing an indirect link: no major knowledge loss) or linking 'vertically' into another subtree (suggesting the node 'belongs' in two different subtrees: a loss of information, magnitude dependent on the information content). In the second case, there is only a problem with loss of knowledge if the deleted arc has an information content that is relatively large in comparison to the globally retained arc. In conclusion, the laboratory case study confirmed that the flexibility of the stochastic tool provided realistic benefits to the designers in terms of selecting options, and that the machine-based knowledge acquisition process was successful. The designer interaction aspects were overcome in this case due to the simplicity of the design problem and the ability of novice designers to accept 'awkward' design tools.

In the industrial case study, the domain experts could only afford to spend the relatively short time required to extract a suitable design database. Further, the induction approach adds the risk of either inducing extrapolation errors or limiting designers to only search within the boundaries set by the set of designs used to train or fit the underlying model. In the industrial case study it was not possible to determine if this was the case. Other important challenges with the underlying model are: the computational efficiency, the level of guidance to select appropriate model tuning parameters, and the ability to assess the model accuracy (Martin and Simpson, 2005). The benefit of the stochastic support tool is that, computationally, it is quite efficient. However, the implementation of the algorithm using standard office tools to ensure wide access, resulted in a barely acceptable refresh rate when the design state was updated. This highlighted the importance of good user interface design for successful decision support tools.

Support tools for the early phases of the design process are by their nature approximations of the actual design domain. These tools are intended to provide a rapid and efficient means for broadly searching the design domain. However, it is important to consider how the model accuracy is being traded off against the search efficiency (Simpson et al., 2001). Clearly the underlying model needs a sufficient level of accuracy. However, it is also important the tool is perceived as providing an accurate representation of the domain. Without this, designers will not trust the tool and hence not use it. A designer obtains trust in a support tool when the underlying model provides similar or tractable predictions to queries that the designer 'knows' the answer to. Effectively, this illustrates to the designer that the underlying model is aligned with their own understanding of the domain. In the laboratory case, there were two aspects that resulted in effective use of the stochastic tool. Firstly, the domain was intuitive and hence it was possible to see that the stochastic tool was providing plausible guidance. Secondly, the subjects had no formal background in the domain, and were therefore more receptive to the guidance being provided. This contrasts with the industrial case study, where the sub-space domain models did not appear to be aligned with the domain experts' views or search methods. The data appears to support a different view on the design domain than that held by the domain experts, however, the experts did not have sufficient time to be able to consider these alternative views more deeply. As a result, the stochastic models were perceived to be inaccurate in some respects.

The final challenge affecting a design support tool is its accessibility. User interface design represents a modern challenge, which is typically case based driven rather than either analytic or deductive (Kim and Yoon, 2005). Szewczyk (2003) reports the need to design the user interface to reflect the manner in which the users work. Therefore it is essential to obtain a good understanding of the users' working practises before designing the user interface. This was not possible in the industrial trial, as access to the domain experts was limited. This partly explains why the design team were not able to provide a critical review of the support tool. In addition, Hauck and Wesiband (2002) report that novice users more readily adapt to using novel support tools than experts. This is supported by the evidence from the two empirical trials. The student designers took to the support tool swiftly as they had never been exposed to a parametric design situation before. Conversely, the expert industrial designers were accustomed to their design tools, and found the stochastic support tool awkward. Ultimately, good user interface design requires that the tool 'leads, follows, and gets out of the way' of the user (Kamper, 2002). The stochastic design support tool fulfils the first two criteria: it leads the designer to a design solution based on the specification provided and it interactively follows the designer updating the design, but it does not get out of the way to allow the user to easily finish the design task.

6 Conclusion

The research in stochastic design support tools has developed, implemented and empirically tested a generic design support tool. The support tool uses prior design databases to learn both the model structure and conditional PDFs between causally related variables. The two empirical trials did highlight a number of challenges relating to the use of such a design support tool.

A number of challenges remain before this type of support tool can be widely adopted. The laboratory empirical trial suggested that the model structure learning algorithm is acceptably accurate. However, the industrial trial could neither confirm or reject the stochastic model structure. There is a strong possibility that this was due to applying the machine learning algorithm on a single dataset where this should have been distinct datasets, or that the machine learning should have been applied to a dataset that was more closely related to the data that the designer directly manipulates. There were both expected and unexpected aspects within the industrial model. Due to commercial pressures, the domain experts were unable to investigate the unexpected model aspects in sufficient detail to either explain or reject these phenomena the stochastic structure learning algorithm identified. These open questions resulted in a degree of uncertainty in the stochastic model quality.

With the more complex industrial design

domain, it became clear that there were also barriers to adopting the support tool created by the nature of the user interface. The support tool was implemented using standard office tools to ensure wide access. However, these tools do not provide a suitable set of user interface options. It was therefore difficult to provide a natural or intuitive interface to the underlying model. Further, it was difficult to clearly visualise the PDFs in the larger industrial case study. These issues were not so great in the simpler laboratory case study, as the complexity of the design domain could still be readily digested by the designers. From the industrial case study, it became clear that the design variables needed to be grouped according to their respective sub-domain models and the designer heuristic of first setting 'narrow mode' PDFs needed to be encoded so as to identify explicitly the suggested variable determination order. In summary, the user interface to the stochastic model required a greater degree of alignment to the design domain. This requires a greater understanding of the domain-context design process. There is also a clear requirement that the designers need time to be able to familiarise themselves with the stochastic data format.

However, the laboratory case study did provide evidence of an emergent design space search method. Subjects using the stochastic tool were observed searching the design space more widely by inspecting a series of PDFs while trying various design variable settings. This emergent behaviour illustrates that the stochastic design support tool provides a novel and powerful means for searching the design space.

In summary, the benefits that were observed from the stochastic design tool were: (1) the ability to flexibly choose among possible design options and being provided with the probability of success for that outcome, (2) the emergence of a novel search strategy, and (3) the automatic knowledge acquisition method. However, there remain significant challenges: (1) the BBN generated through the machine learning process must be able to be verified by domain experts for them to trust the support system, and (2) the interaction between designer and support system must be aligned with the design process.

Further work is required to address these hurdles. This will need to consider both the user interface aspects of the tool and the means for training designers in the use of the tool. This will require a related line of research to investigate human abilities to reason with stochastic, and in particular Bayesian, domain representations: what and how is the most intuitive manner for presenting these multiple options and how is this information processed by the designer. In addition, a modular methodology for developing interfaces from this generic design support tool with more specific design domains and tools. By providing a lightweight domain specific interface, this will enable the adoption of stochastic design representation across a wide collection of design domains. So long as these hurdles remain, the adoption of stochastic approaches to support the design process will remain limited.

Acknowledgements

This research is funded by a Nuffield Foundation Award to Newly Appointed Lecturers in Science, Engineering and Mathematics (Grant number: NAL/00846/G). Thanks must also be given to the design students and designers from Rolls-Royce (Aerospace) who provided their time to support the empirical work. Thanks is also given to the anonymous reviewers whose inputs vastly improved the final presentation of this paper.

References

Apel, H., Thieken, A. H., Merz, B. and Bloschl, G. (2006). A probabilistic modelling system for assessing flood risks, *Natural Hazards* **38**(1–2): 79–100.

Blake, C. L. and Merz, C. J. (1998). UCI repository of machine learning databases. `http://www.ics.uci.edu/~mlearn/MLRepository.html`.

Chen, J., Greiner, R., Kelly, J., Bell, D. and Liu, W. (2002). Learning bayesian networks from data: An information-theory based approach, *Artificial Intelligence* **137**(1–2): 43–90.

Chen, V. C. P., Tsui, K. L., Barton, R. R. and Meckesheimer, M. (2006). A review on design, modelling and applications of computer experiments, *IIE Transactions* **38**(4): 273–291.

Chong, Y., Chen, C.-H. and Leong, K. (2009). A heuristic-based approach to conceptual design, *Research in Engineering Design*.

Clarkson, P. J., Simons, C. and Eckert, C. M. (2004). Predicting change propagation in complex design, *Journal of Mechanical Design* **126**(5): 788–797.

Cross, N. (1994). *Engineering Design Methods: Strategies for Product Design*, John Wiley, Chichester, UK.

Eliashberg, J., Jonker, J. J., Sawhney, M. S. and Wierenga, B. (2000). MOVIEMOD: An implementable decision-support system for prerelease market evaluation of motion pictures, *Marketing Science* **19**(3): 226–243.

Fernandez, M. G., Seepersad, C. C., Rosen, D. W., Allen, J. K. and Mistree, F. (2005). Decision support in concurrent engineering — the utility-based selection decision support problem, *Concurrent Engineering: Research and Applications* **13**(1): 13–27.

Forouraghi, B. (1999). On utility of inductive learning in multiobjective design, *Artificial Intelligence for Engineering Design Analysis and Manufacturing* **13**(1): 27–36.

Gaines, B. R. and Shaw, M. L. G. (1993). Eliciting knowledge and transferring it effectively to a knowledge-based system, *IEEE Transactions on Knowledge and Data Engineering* **5**(1): 4–14.

Geyer, P. (2008). Multidisciplinary grammars supporting design optimization of buildings, *Research in Engineering Design* **18**(4): 197–216.

Hauck, R. V. and Wesiband, S. (2002). When a better interface and easy navigation aren't enough: Examining the information architecture in a law enforcement agency, *Journal of the American Society for Information Science as Technology* **53**(10): 846–854.

He, L., Chan, C. W., Huang, G. H. and Zeng, G. M. (2006). A probabilistic reasoning-based decision support system for selection of remediation technologies for petroleum-contaminated sites, *Expert Systems with Applications* **30**(4): 783 – 795.

Hollins, B. and Pugh, S. (1990). *Successful Product Design*, Butterworths.

Hughes, E. J., Richards, T. G. and Tilley, D. G. (2001). Development of a design support tool for fluid power system design, *Journal of Engineering Design* **12**(2): 75–92.

Kamper, R. J. (2002). Extending the usability heuristics fir desugb abd evaluation: Lead, follow get out the way, *International Journal of Human-Computer Interaction* **14**(3–4): 447–462.

Kim, H. and Yoon, Y. C. (2005). Supporting the cognitive process of user interface design with resusable design cases, *International Journal of Human-Computer Studies* **62**(4): 457–486.

Mahdavi, A. and Gurtekin, B. (2004). Generating the design-performance space via simulation and machine learning, *Journal of Architectural and Planning Research* **21**(4): 350–362.

Maher, M. L. and Gómez de Silva Garza, A. (1997). Case-based reasoning in design, *IEEE Expert: Intelligent Systems and their Applications* **12**(2): 34–41.

Martin, J. D. and Simpson, T. W. (2005). Use of kriging models to approximate deterministic computer models, *AIAA Journal* **43**(4): 853–863.

Matthews, P. C. (2006). Bayesian networks for design: A stochastic design search method, *in* J. S. Gero (ed.), *Proceeding of the Design Computing and Cognition conference*, Kluwer, Eindhoven. (in press).

Matthews, P. C. (2008). A Bayesian support tool for morphological design, *Advanced Engineering Informatics* **22**(2): 236–253.

Matthews, P. C., Standingford, D. W. F., Holden, C. M. E. and Wallace, K. M. (2006). Learning inexpensive parametric design models using an augmented genetic programming technique, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* **20**(1): 1–18.

Moore, C. J., Miles, J. C. and Ress, D. W. G. (1997). Decision support for conceptual bridge design, *Artificial Intelligence in Engineering* **11**(3): 259–272.

Ong, Y. S. and Keane, A. J. (2002). A domain knowledge based search advisor for design problem solving environments, *Engineering Applications of Artificial Intelligence* **15**(1): 105–116.

Osio, I. G. and Amon, C. H. (1996). An engineering design methodology with multistage Bayesian surrogates and optimal sampling, *Research in Engineering Design* **8**(4): 189–206.

Pacheco, J. E., Amon, C. H. and Finger, S. (2003). Bayesian surrogates applied to conceptual stages of the engineering design process, *Journal of Mechanical Design* **125**(4): 664–672.

Pahl, G. and Beitz, W. (1996). *Engineering Design: A Systematic Approach*, second edn, Springer-Verlag London.

Pearl, J. (1995). Causal diagrams for empirical research, *Biometrika* **82**(4): 669–709.

Pearl, J. and Verma, T. S. (1991). A theory of inferred causation, *in* J. F. Allen, R. Fikes and E. Sandewall (eds), *KR'91: Principles of Knowledge Representation and Reasoning*, Morgan Kaufmann, San Mateo, California, pp. 441–452.

Pons, D. J. and Raine, J. K. (2004). Design with uncertain qualitative variables under imperfect knowledge, *Proceedings of the Institution of Mechanical Engineers Part B: Journal of Engineering Manufacture* **218**: 977–986.

Pons, D. J. and Raine, J. K. (2005). Simulation of key performance characteristics under uncertainty, *Proceedings of the Institution of Mechanical Engineers Part B: Journal of Engineering Manufacture* **219**(1): 151–162.

Potter, S., Culley, S. J., Darlington, M. J. and Chawdhry, P. K. (2003). Automatic conceptual design using experience-derived heuristics, *Research in Engineering Design* **14**(3): 131–144.

Potter, S., Darlington, M. J., Culley, S. J. and Chawdry, P. K. (2001). Design synthesis knowledge and inductive machine learning, *Artificial Intelligence for Engineering Design Analysis and Manufacturing* **15**(3): 233–249.

Resnick, M. L. and Vaughan, M. W. (2006). Best practices and future visions for search user interfaces, *Journal of the American Society for Information Science and Technology* **57**(6): 781–787.

Rinkus, S., Walji, M., Johnson-Troop, K. A., Malin, J. T., Turley, J. P. and Zhang, J. J. (2005). Human-centered design of a distributed knowledge management system, *Journal of Biomedical Informatics* **38**(1): 4–17.

Rivard, H. and Fenves, S. J. (2000). SEED-Config: A case-based reasoning system for conceptual building design, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* **14**(5): 415–430.

Sahinidis, N. V. (2004). Optimization under uncertainty: State-of-the art and opportunities, *Computers & Chemical Engineering* **28**(6–7): 971–983.

Scwabacher, M., Ellman, T. and Hirsh, H. (1998). Learning to set up numerical optimizations of engineering designs, *Atrificial Intelligence for Engineering Design Analysis and Manufacturing* **12**(2): 173–192.

Simpson, T. W., Peplinski, J. D., Koch, P. N. and Allen, J. K. (2001). Metamodels for computer-based engineering design: Survey and recommendations, *Engineering with Computers* **17**(2): 129–150.

Szewczyk, J. (2003). Difficulties with the novices' comprehension of the computer aided design (CAD) interface: Understanding visual representations of CAD tools, *Journal of Engineering Design* **14**(2): 169–185.

Ulrich, K. T. and Eppinger, S. D. (2000). *Product Design and Development*, second edn, McGraw-Hill, Boston, MA.

Zhang, W., Tor, S. and Britton, G. (2002). Automated functional design of engineering systems, *Journal of Intelligent Manufacturing* **13**(2): 119–133.

Ziv-Av, A. and Reich, Y. (2005). SOS — subjective objective system for generating optimal product concepts, *Design Studies* **26**(5): 509–533.

Zwicky, F. (1967). The morphological approach to discovery, invention, research and construction, *in* F. Zwicky and A. G. Wilson (eds), *New Methods of Thought and Procedure*, Springer-Verlag.